所属类别	2022年“华数杯”全国大学生数学建模竞赛	参赛编号
本科		CM2205540

基于混合整数线性规划的水下机器人组装计划求解

摘要

可在水下移动、具有视觉和感知系统的自来水管道清理机器人（Water pipe cleaning robot，简称WPCR）能通过遥控或自主操作方式、使用机械臂代替或辅助人去完成自来水管道垃圾清理任务。这种装置能及时清理管道，既可提高自来水的品质，也能够保证水流畅通。该产品正受到水务公司和家庭住户的青睐，因此，对该装置的生产规划问题应运而生。该问题是一个多级生产与库存管理的问题，需要合理安排每日生产计划并使库存得到规划才能让此问题得到很好的解决。

问题1：首先根据题目信息可知在给定期限一周内开始阶段无任何存货，且期限结束后也不遗留任何组件库存，那么每日生产计划需得到合理安排刚好用完。当日采购的零件马上就可以用于组装，组装出来的部件也可以马上用于当日组装成品装置WPCR。我们据此建立了目标函数即为每日需要生产的产品生产准备费用与每日剩余库存费用之和，赋予生产部件数量、是否生产组件、组件库存数量等为决策变量，构建物流平衡方程：每个时段前一时段库存加当前时段生产量，减去该项目当前时段的需求量和用于组装其他项目的量等于当前时段库存。决策目标为总成本，给定约束条件，建立优化模型求解,得出最小成本为6260.9元

问题2：问题2提供了新的条件，即当日需要生产的项目的部件需要提前一天采购或组装。周一开始时的存货为周日库存，周日结束时同样需要给下周一的生产提供库存准备。据题我们可将目标生产计划理解为一个周期闭环，周一和周日变得连续，我们在问题1的基础上添加了新的约束:每天组装所消耗的配件数量不大于前一天的库存数量，新增约束后,物流平衡方程无需改变,便可求出最优解,得出的最小成本为177212.5元。

问题3：问题3在问题2的基础上添加的新的条件，即210天内需要进行7天的维修，在维修当天工厂无法进行任何组件的生产，同时任意两次维修的日期必须要间隔6天以上，在每次维修过后的5天，工厂的生产工时会有一定的增加，在将这些约束加入问题 2 的优化模型的同时加入一个新的决策变量来表示是否需要维修即可建立起基于问题3的混合整数线性规划，利用Matlab对于此模型求解，可得到最优解，求解后可知最小成本为5317581元。

问题4：题目给出了前30周的历史周订单数据，我们可以根据前30周的数据，设其需求近似的服从正态分布，以此为样本空间，进行区间估计，求出一周中各天在置信水平为0.95，显著性水平为1-0.95=0.05的单侧置信上限，表示一周内该天WPCR需求落在此区间的概率为95%。对每周总体WPCR需求作为样本空间，进行区间估计，求出每周需求的置信水平为0.85，显著性水平为1-0.85=0.15的单侧置信上限，表示每周的WPCR需求小于此上限的概率为85%，据此制定了新的周生产计划，该计划下最小成本为177942.5元。

关键词：0-1规划、混合线性整数规划、区间估计、正态分布

一、问题重述

自来水管道清理机器人（Water pipe cleaning robot，简称 WPCR）是一种可在水下移 动、具有视觉和感知系统、通过遥控或自主操作方式、使用机械臂代替或辅助人去完成 自来水管道垃圾清理任务的装置。运用这种装置能够及时清理管道，既可提高自来水的 品质，也能够保证水流畅通，因而越来越受到水务公司和家庭住户的青睐。

某工厂生产的 WPCR 装置需要用 3 个容器艇(用 A 表示)、4 个机器臂(用 B 表示)、5 个动力系统（用 C 表示）组装而成。每个容器艇（A）由 6 个控制器（A1）、8 个划桨 （A2）和 2 个感知器（A3）组成。每个机器臂（B）组成比较复杂，简单可划分为 2 个 力臂组件（B1）和 4 个遥感器（B2）组成。每个动力系统（C）由 8 个蓄电池（C1）、2 个微型发电机（C2）和 12 个发电螺旋（C3）组成。也就是说组装一个完整的 WPCR 装置，需要 3 个容器艇（A），包括 18 个控制器（A1）、24 个划桨（A2）以及 6 个感知器（A3）。组装一台 WPCR 需要的其他部件数以此类推。组装 WPCR 所需要的产品统 称为组件，包括 A 和 A1、A2、A3，B 和 B1、B2，C 和 C1、C2、C3。

该工厂每次生产计划的计划期为一周（即每次按照每周 7 天的订购数量实行订单生 产），只有最终产品 WPCR 有外部需求，其他组件不对外销售。容器艇（A）、机器臂（B）、动力系统（C）生产要占用该工厂最为关键的设备，因而严格控制总生产工时。A、B、C 的工时消耗分别为 3 时/件、5 时/件和 5 时/件，即生产 1 件 A 需要占用 3 个工 时，生产 1 件 B 需要占用 5 个个工时，生产 1 件 C 需要占用 5 个工时。每天的 WPCR 外部 需求数及关键设备总工时限制见表 5。

为了顺利生产 WPCR，工厂在某一天生产组件产品时，需要付出一个与生产数量无关的固定成本，称为生产准备费用。比如第一天生产了 A，则要支付 A 的生产准备费用，若第二天再生产 A，则需要再支付 A 的生产准备费用。如果某一天结束时某组件有库存存在，则工厂必须付出一定的库存费用（与库存数量成正比）。数据见表 6。另外，按照工厂的信誉要求，目前接收的所有订单到期必须全部交货，轻易不能有缺货事件发生。

问题 1：若该工厂第一天（周一）开始时没有任何组件库存，也不希望第 7 天（周日）结 束后留下任何组件库存。每天采购的组件马上就可用于组装，组装出来的组件也可以马 上用于当天组装成 WPCR。若要求总成本最小，请问如何制定每周 7 天的生产计划？

问题 2：然而，事实上，组件 A、B、C 需要提前一天生产入库才能组装 WPCR，A1、A2、A3、B1、B2、C1、C2、C3 也需要提前一天生产入库才能组装 A、B、C。在连续多周生产情况下，需要统筹规划。比如在周一生产 WPCR 前一天（上周周日）必须事先准备好组件库存，而且在本周日必须留下必要的组件库存用以保障下周一的生产。每周的 WPCR 需求和关键设备工时限制以及每次生产准备费用和单件库存费用数据见表 5、表 6，请问如何制定每周 7 天的生产计划以求总成本最低？

问题 3：为了保障生产的持续性，工厂需要在 30 周 210 天里必须设置 7 次停工检修，每次检修时间为 1 天。检修之后关键设备生产能力有所提高，检修后的第一天 A、B、C 生产总工时限制将会放宽 10%，随后逐日减少放宽 2%的比例，直至为 0（如第一天放宽 10%，第二天就放宽 8%，...）。检修日的订单只能提前安排生产，当天不能生产任何组件。假设每周的关键设备工时限制以及每次生产准备费用和单件库存费用数据不变，任意两次检修之间要相隔 6 天以上，请问，检修日放在哪几天最为合适（总成本最小）？

问题 4：在生产实际中，在未知 WPCR 外部需求订单的前提下，公司需要有一个稳妥

的单 周生产计划。接问题 2，表 7 数据视为历史周订单数据，在不知未来某周 7 天订单数且 继续追求周总成本最小的前提下，如何制定周生产计划，既能够保障每天的WPCR 订单 均以 95%以上的概率保证正常交付，又能够以 85%以上的概率保证整周的WPCR 订单能 正常交付？

二、问题假设

1.假设组装产品 WPCR、采购小部件（如A_1、B_1）不消耗时间，只有关键设备 A、B、C 的组装需要消耗工时。
2.假设总成本仅含生产准备费用及库存费用。
3.假设有关原材料、人力和能量等直接成本作为常数不予考虑。
4.假设检修间隔六天以上实际意义为若周一检修，那么周日也可以检修。
5.假设问题三条件中第 1 天的初始存货为第 210 天库存。
6.假设 30 周历史订单周一至周日 WPCR 需求服从正态分布。

三、符号说明

符号	符号表示含义
$\boldsymbol{I_{i,t}}$	每种组件每日库存数量
$\boldsymbol{X_{i,t}}$	每种组件每日生产数量
$\boldsymbol{Y_{i,t}}$	每种组件当日是否生产
$\boldsymbol{F}$	总成本
$\boldsymbol{S_i}$	每种组件生产准备费用
$\boldsymbol{H_i}$	每种组件单件库存费用
$\boldsymbol{D_{i,t}}$	各组件每日需求量
$\boldsymbol{R_{i,s(i)}}$	存储不同组件所需后继组件数量
$\boldsymbol{C_i}$	各部件消耗工时
$\boldsymbol{L_t}$	每日总工时限制
M	足够大的实数
$\boldsymbol{W_t}$	当日是否检修
$\boldsymbol{mu_i}$	单侧置信上限
$\bar{\mathrm{a}}_{\mathrm{i}}$	样本均值
s	标准差
$\boldsymbol{t_\alpha(29)}$	t 分布表中随测点数(29)和置信度(α)而变的系数

四、问题分析

4.1 问题一的分析

该题目需考虑在有限计划时间内，给定产品在期限内的外部需求及产品生产费用和相关库存费用及关键设备工时限制后，确定每一生产项目的期限内的生产批量。

问题给定期限一周，假设了周一开始时及周日结束后均无任何组件库存，且采购部件、组装 WPCR 均不消耗时间，仅组装设备 A、B、C 需消耗工时。每天采购组件及组装 WPCR 均可在当日完成。

为使总成本费用最小，我们使在期限一周内的 WPCR 产品的总产量刚好等于产品总需求量。考虑到产品 A、B、C 及 WPCR 的生产准备费用及库存费用的差异，我们将每日是否生产 A、B、C 及 WPCR、每天生产部件数量、每种组件在每天的库存数量设定为决策变量。目标函数即为每一部件的总生产准备费用和总库存费用之和，我们在外加每日外部需求和每日产品工时限制等约束条件下，便可以建立问题一的优化模型。

4.2 问题二的分析

问题二在问题一的基础上增加了一些实际生产中所要考虑的情况。组件 A、B、C 需要提前一天生产入库才能组装 WPCR，A1、A2、A3、B1、B2、C1、C2、C3 也需要提前一天生产入库才能组装 A、B、C。在进行连续多周的生产时，要进行必要的规划设计。可以将周日与周一连接，前日的组件库存必须满足今日的生产消耗，同理今日的组件库存也必须满足明日的生产消耗。因此不同于问题一，周一若要生产 WPCR 或 A,B,C 时，前一天必须库存所需的原料组件，周一开始时就会存在有前一天库存的情况，在周日也会存在库存的情况。对于问题一的决策变量同样适用于问题二，目标函数同样表示总生产准备费用和总库存费用之和。在问题一的约束条件下，另加当日组件生产所需原料的前一日的库存要求，可进一步建立问题二的优化模型。

4.3 问题三的分析

问题三是在问题二的基础之上将总的生产时间从一周变为了 210 天，同时增加了需要对仪器设备进行检修的约束条件，同时在维修之后五天的生产工时限制也会随之而改变，另外：检修日当天无法生产任何组件以及任意两次维修时间必须间隔 6 天以上，210 天内的维修次数为 7 次。相较于问题二，问题三只是对约束条件做出了一定地增加，由于目标依旧是让总成本最小，所以目标函数不需要改变，同时因为我们需要知道具体的检修时间，所以我们在问题二原有的决策变量的基础之上增加任意一天是否需要检修这一决策变量。对于问题三的约束条件，由于问题三是建立于问题二之上，所以我们只需要在对问题二原有的约束条件之上做出一点小小的改变的同时，加上关于检修相关的几个约束条件，即可建立问题三的优化模型。

4.4 问题四的分析

问题四同为问题二的改进问题，在未知 WPCR 外部需求订单的前提下，需要制定一项稳妥的单周生产计划，需要既能够保障每天的 WPCR 订单均以 95%以上的概率保证正常交付，又能够以 85%以上的概率保证整周的 WPCR 订单能正常交付，并且考虑到使总成本最小，在此条件下给出单周生产计划。题目给出了前 30 周的历史周订单数据，

我们可以根据前 30 周的数据，设其需求近似的服从正态分布，以此为样本空间，进行区间估计，求出一周中各天在置信水平为 0.95，显著性水平为 1-0.95=0.05 的单侧置信上限，表示一周内该天 WPCR 需求落在此区间的概率为 95%。对每周总体 WPCR 需求作为样本空间，进行区间估计，求出每周需求的置信水平为 0.85，显著性水平为 1-0.85=0.15 的单侧置信上限，表示每周的 WPCR 需求小于此上限的概率为 85%。以此上限来确定生产计划中的 WPCR 需求，能够满足既能够保障每天的 WPCR 订单均以 95%以上的概率保证正常交付，又能够以 85%以上的概率保证整周的 WPCR 订单能正常交付的要求。再将此需求与问题二求解的优化模型进行结合，就可保证在满足订单交付要求的条件下求出使总费用最低的单周生产计划。

五、问题一模型的建立与求解

5.1 模型准备

首先我们对该问题进行分析，根据参考文献[1]可知，本问题是一个有瓶颈设备多级生产问题，目标产品 WPCR 装置是由容器艇（A）、机械臂（B）、动力系统（C）等组件组装而成。“多级”，指的是需要考虑产品是通过多个生产阶段(工艺)生产出来的。主要任务是通过组装生产产品 WPCR，用于满足外部市场需求。于是，我们可以建立 WPCR 装置的构成与组装过程如下图：

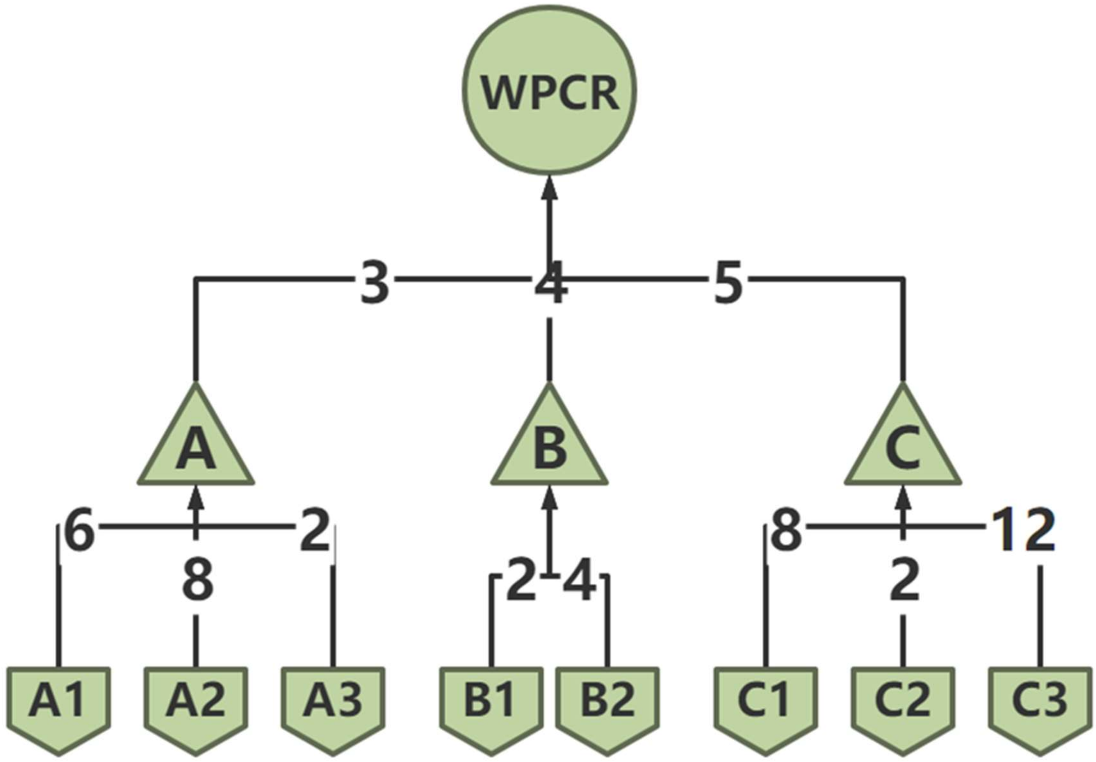

图 1 WPCR 装置生产流程

流程图中绿色图标内字符表示最终产品 WPCR 与生产它所必须的各种组件，连线表示下层的组件组成上层组件，连线上的数值表示上层组件的生产需要下层组件的数目。即组成一个 WPCR 需要 3 个 A，4 个 B，5 个 C；而组成一个 A 需要 6 个 A1，8 个 A2，2 个 A3；组成一个 B 需要 2 个 B1，4 个 B2；组成一个 C 需要 8 个 C1，2 个 C2，12 个 C3.。

给定计划期限为 1 周，最终只有产品 WPCR 装置有外部需求，即产品 WPCR 需每天满足需求量。每天 WPCR 装置需求及关键设备工时限制如下：

表 1 WPCR 需求和关键设备工时限制

周次	1	2	3	4	5	6	7
WPCR 需求数（个）	39	36	38	40	37	33	40
A、B、C 生产工时限制（工时）	4500	2500	2750	2100	2500	2750	1500

合理安排每日库存也是使总成本达到最小的重要因素之一，针对问题一，每天需要生产的组件均可当日购买并组装，组件 A、B、C 也可在当日组装成装置 WPCR。由表二生产准备费用和单件库存费用可知，组装好的产品库存费用明显低于未组装好的组件库存费用。此外，生产准备费用为当日生产产品所需的固定费用，经表比较产品的生产准备费用与单间库存费用，发现在一定的数量限制下，产品的库存费用可以低于同一产品的生产准备费用，这为后续的求解最优目标函数提供了思路。

表 2 生产准备费用和单件库存费用（单位：元）

产品	WPCR	A	A1	A2	A3	B	B1	B2	C	C1	C2	C3
生产准备费用	240	120	40	60	50	160	80	100	180	60	40	70
单间库存费用	5	2	5	3	6	1.5	4	5	1.7	3	2	3

因为每一生产项目若在当天生产时必须经过生产准备，需要给与一次性的生产准备费用，在结合每天的工时限制，每天多生产的项目将会产生库存费用，在该题我们假设总费用仅为生产准备费用和库存费用，其余有关原材料、人力和能量等直接成本作为一个常数不予考虑。

5.2 模型建立

为使总成本最小，期限内装置总产量一定恰好等于装置总需求量。在满足每日工时限制的情况下，我们的问题目标即是令生产准备费用和库存费用总和最小。目标函数应为每日各项目生产准备费用和每日库存费用总和，即：

决策变量为：

$$X_{i,t} \quad Y_{i,t} \quad I_{i,t}$$

目标函数为：

$$\min \ F = \sum_{i=1}^{12} \sum_{t=1}^{7} \left(S_i Y_{i,t} + H_i I_{i,t} \right)$$

约束条件有：

1. 生产守恒约束:
   前一天库存+今日生产-今日库存=今日需求+今日生产后继产品所需数量
2. 生产工时约束:
   每天生产的各个组件*该组件所消耗工时≤当天生产工时限制
3. 每日各个组件产量约束:
   用一个足够大的实数 M(5e4)来约束每天各个组件的产量
4. 基本约束条件
   每日生产组件、每日库存组件为≥0 的整数；每日是否生产为 0，1 变量

$$\begin{cases} i = 1,2,\ldots 12\ ;\ t = 1,2,..7 \\ \begin{cases} I_{i,t-1} + X_{i,t} - I_{i,t} = D_{i,t} + R_{i,s(i)} * X_{s(i),t} \\ \sum_{i=1}^{12} C_i * X_{i,t} \le L_t \\ 0 \le X_{i,t} \le M * Y_{i,t} \end{cases} \\ M(\text{足够大的实数}) \\ Y_{i,t} \in 0|1 \qquad I_{i,t} \ge 0 \qquad X_{i,t}\text{，} I_{i,t} \in Z \end{cases}$$

5.3 问题求解

根据以上建立的数学模型，我们运用数据处理软件 MatLab 编写了本题程序进行求解，建立了物流平衡方程及产量约束不等式对目标函数进行约束（程序见本文附件 1），代码实现目的并运行后，求得最优解如下：

表 3 问题一生产规划

周次/项目	WPCR 组装数量	A 组装数量	B 组装数量	C 组装数量	生产准备费用	库存费用
周一	83	249	332	416	1200	221.7
周二	0	0	344	0	340	557.7
周三	81	243	0	404	860	285
周四	0	0	0	0	0	85
周五	48	144	173	240	1200	111.5
周六	51	153	203	255	1200	200
周日	0	0	0	0	0	0
总和	263	789	1052	1315	总成本（元）	6260.9

以上便是我们对目标函数求解得出的最优解，最终成本为 6260.9 元，包括生产准备费用 4800 元及库存费用 1460.9 元，每日组装数量及每日库存数量均已经列出。

对于问题一，因为每天采购的组件马上就可用于组装，组装出的组件也可以马上用于当天组成 WPCR，考虑到采购组件（如 A1、B1 等）库存费用过大，为使总费用最小，该情况下采购组件均不会在当天库存，在当天采购当天组装。则问题一每日库存安排如下：

表 4 问题一库存规划

周次/项目	WPCR 库存数量	A 库存数量	B 库存数量	C 库存数量
周一	44	0	0	1
周二	8	0	344	1
周三	51	0	20	0
周四	11	0	20	0
周五	22	0	1	0
周六	40	0	0	0
周日	0	0	0	0
总和（个）	176	0	385	2

六、问题二模型的建立与求解

6.1 模型准备

相较于问题一，问题二增加了一些实际生产中所要考虑的情况。组件 A、B、C 需要提前一天生产入库才能组装 WPCR，A1、A2、A3、B1、B2、C1、C2、C3 也需要提前一天生产入库才能组装 A、B、C。因此，在连续多周生产的时候，需要进行必要的统筹规划，包括周一的生产原料需要周日准备好，同理周日必须库存周一生产所需要的原料。可以将整周进行首尾衔接。

因此，在问题一建立的模型基础下，我们需要新增约束条件，包括每天生产的产品或组件必须受到前一天该组件的原料组件库存数量的约束，并且还应当在当日库存明日所需的原料部件。其余的约束同问题一一致，以此来通过线性规划的经典解法求出最优解。

6.2 模型建立

我们的求解目标仍然是解出目标函数的最小值，目标函数由各决策变量与费用之间的关系来确定，使得组成目标函数的准备费用与库存费用达到最小，并且满足一定的约束条件。

决策变量为：

$$X_{i,t} \quad Y_{i,t} \quad I_{i,t}$$

目标函数为：

$$\min \ F = \sum_{i=1}^{12} \sum_{t=1}^{7} \left(S_i Y_{i,t} + H_i I_{i,t} \right)$$

约束条件有：

1. 问题一约束
2. 新增生产与前日库存之间的约束：

   今日生产某组件消耗的原料组件数目≤前日相应原料组件库存；

   周日各组件库存≤一周内每日各组件库存之和

即

$$
\begin{cases}
i = 1,2,\ldots 12\ ;\ t = 1,2,..7 \\
\begin{cases}
I_{i,t-1} + X_{i,t} - I_{i,t} = D_{i,t} + R_{i,s\{i\}} * X_{s\{i\},t} \\
\sum_{i=1}^{12} C_i * X_{i,t} \le L_t \\
0 \le X_{i,t} \le M * Y_{i,t}
\end{cases} \\
\begin{cases}
I_{i,7} \le \sum_{t=1}^{7} X_{i,t} \\
R_{i,s\{i\}} * X_{s\{i\},t} \le I_{i,t-1} \\
Y_{i,t} \in 0|1 \qquad I_{i,t} \ge 0 \qquad X_{i,t},\ I_{i,t} \in Z
\end{cases}
\end{cases}
$$

6.3 模型求解

同样，我们根据修改后建立的数学模型，利用 Matlab 软件编程对该问题进行了求解，相较于问题 1，增添约束：每天组装所消耗的配件数量不大于前一天的库存数量后，求出了新的最优解如下表 5：

表 5 问题二生产规划

周次/项目	WPCR 组装数量	A 组装数量	B 组装数量	C 组装数量	生产准备费用	库存费用
周一	0	246	341	411	630	33737.7
周二	82	0	0	500	750	31296.7
周三	0	300	370	0	280	2101.7
周四	100	0	0	0	420	9899.7
周五	0	0	341	0	480	42572.2
周六	0	243	0	404	300	1686
周日	81	0	0	0	740	52318.5
总和	263	789	1052	1315	总成本（元）	177212.5

问题 2 我们经过 Matlab 软件代码运行求出最终最优解为总成本 177212.5 元，其中含生产准备费用 3600 元及库存费用 173612.5 元，该库存费用主要为采购部件（如 A1、B1、C1）占用。

该问题与问题 1 不同，每天需要组装的组件需要提前一天准备好，意味着采购部件与相应组装成组件 A、B、C 等组件不能在同一天发生，对于每一部件，每一天的生产准备费用如下表：

表 6 问题二每日生产准备费用

项目/日	周一	周二	周三	周四	周五	周六	周日
WPCR	0	240	0	240	0	0	240
A	120	0	120	0	0	120	0
A1	0	40	0	0	40	0	40
A2	0	60	0	0	60	0	60
A3	0	50	0	0	50	0	50
B	160	0	160	0	160	0	0
B1	0	80	0	80	0	0	80
B2	0	100	0	100	0	0	100
C	180	180	0	0	0	180	0
C1	60	0	0	0	60	0	60
C2	40	0	0	0	40	0	40
C3	70	0	0	0	70	0	70
总费用	630	750	280	420	480	300	740

问题 2 的采购部件因为需要提前一天购买，又因为采购部件占据着高昂的库存费用，总成本因采购部件的占据将大大提高，每一部件每天库存情况及费用如下表：

表 7 问题二库存规划

项目数	周一	周二	周三	周四	周五	周六	周日
WPCR	2	48	10	70	33	0	41
A	246	0	300	0	0	243	0
A1	0	1800	0	0	1458	0	1476
A2	0	2400	0	0	1944	0	1968
A3	0	600	0	0	486	0	492
B	358	30	400	0	341	341	17
B1	0	740	0	682	0	0	682
B2	0	1480	0	1364	0	0	1364
C	411	501	501	1	1	405	0
C1	4000	0	0	0	3232	0	3288
C2	1000	0	0	0	808	0	822
C3	6000	0	0	0	4848	0	4932
总费用	33737.7	31296.7	2101.7	9899.7	42572.2	1686	52318.5

七、问题三模型的建立与求解

7.1 模型准备

问题三从实质上来说是在问题二的基础之上加上了一定的条件而产生，所以我们只需要在已经建立的问题二优化模型基础上加上一些因素，同时将将一些原有因素变量化即可。

对于问题三。我们可以知道是在问题二的基础之上，增加了需要在 210 天内安排 7 次检修这个条件，同时，在每一次维修过后的五天，生产能力都会得到一定的提升，从而提升了这五天的生产时间上限，在这个条件之下，我们需要注意的是任意两次的维修需要间隔 6 天以上，另外在维修当天工厂不能够进行任何组件的生产。将这些新增条件化作求解问题所需的约束条件，同时增加任意一天是否需要对工厂进行检修这一决策变量即可实现基于问题三的混合整数线性优化模型的建立。

7.2 模型建立

根据问题三分析结果，问题三在问题二的基础之上进行更新条件，因此问题二的约束条件依然适用。并做出相应改动，将 30 周 210 天看作一个循环体，需要将问题一、二中的一周天数限制改为30周210天限制，还需要引入一个当日是否检修的决策变量，将检修后的工时放宽比例设置为常量建立模型。因此，在继续使用线性规划经典求解时，目标函数以及约束都要做出相应的改变。

决策变量（增加W_t):

$$X_{i,t} \quad Y_{i,t} \quad I_{i,t} \quad W_t$$

目标函数为：

$$\min \ F=\sum_{i=1}^{12}\sum_{t=1}^{210}\left(S_iY_{i,t}+H_iI_{i,t}\right)$$

约束条件为：

1. 问题二约束条件（改动工时限制及周期天数）
2. 新增约束：

   单次循环时间变为 210 天；

   若当天检修，后一天起工时放宽一定比例；

   若当天不检修，当天工时限制同问题二工时限制；

   在六天内检修天数不超过一天，总检修天数=7；

即

$$
\begin{cases}
i = 1,2,\ldots 12\ ;\ t = 1,2,..210 \\
\begin{cases} I_{i,t-1} + X_{i,t} - I_{i,t} = D_{i,t} + R_{i,s\{i\}} * X_{s\{i\},t} \\ 0 \le X_{i,t} \le M * Y_{i,t} \end{cases} \\
\begin{cases} I_{i,7} \le \sum_{t=1}^{7} X_{i,t} \\ R_{i,s\{i\}} * X_{s\{i\},t} \le I_{i,t-1} \end{cases} \\
\begin{cases} if \quad W_t == 0 \qquad \text{——检修，改变工时约束} \\ X_{i,t} = 0 \\ \sum_{i=1}^{12} C_i * X_{i,t+1} \le 1.1 * L_{t+1} \\ \sum_{i=1}^{12} C_i * X_{i,t+2} \le 1.08 * L_{t+2} \\ \sum_{i=1}^{12} C_i * X_{i,t+3} \le 1.06 * L_{t+3} \\ \sum_{i=1}^{12} C_i * X_{i,t+4} \le 1.04 * L_{t+4} \\ \sum_{i=1}^{12} C_i * X_{i,t+5} \le 1.02 * L_{t+5} \end{cases} \\
\begin{cases} if \quad W_t == 1 \qquad \text{——不检修，不改变工时约束} \\ \sum_{i=1}^{12} C_i * X_{i,t} \le L_t \end{cases} \\
\begin{cases} 5 \le \sum_{t=1}^{6} W_t \\ \sum_{t=1}^{210} W_t = 203 \end{cases} \\
Y_{i,t} \in 0|1 \qquad I_{i,t} \ge 0 \qquad X_{i,t}\text{，}I_{i,t} \in Z \\
\text{（}W_t\text{为 0 时当天检修，为 1 时不检修）}
\end{cases}
$$

7.3 模型求解

将期限 210 天写入矩阵后，改变原有物流平衡方程，增添新的决策变量，及每天是否生产，若不生产，则该天作为检修日检修。根据以上约束条件修改好代码后，通过 Matlab 软件对目标函数总成本求得了最优解，则检修日期为：

表 8 问题 3 检修日期及总成本

第一次	第二次	第三次	第四次	第五次	第六次	第七次	总成本（元）
第 7 天	第 13 天	第 172 天	第 179 天	第 185 天	第 192 天	第 203 天	5317581

这是我们根据所建立的物流平衡方程及约束条件求解所得（代码可见附录 3），我们的思路为构建矩阵 W，矩阵内元素值为 1 代表当天不检修，值为 0 代表当天检修，并增加新的约束条件：

若当天检修，则之后五天的生产工时限制会增多，意味着当天可以组装更多组件；若当天不检修，则工时限制不变。

根据题目所给条件 30 周 210 天内只能够安排 7 天进行检修，则 W 矩阵元素之和恰好为 203。

每两次检修时间至少要间隔 6 天，这里我们使用了一个循环，构建新的数组，数组内元素值等于任意连续六天的 W 矩阵元素值值和，若和为 5，代表这 6 天内有一天需要检修；若和为 6，代表这 6 天都不需要检修。

结合问题所设决策变量及约束条件，编写好代码，运行并求得了目标函数的最优解：检修日期分别为第 7 天、第 13 天、第 172 天、第 179 天、第 185 天、192 天、第 203 天，最终总成本为 5317581 元。

八、问题四模型的建立与求解

8.1 模型准备

问题四同样在问题二的基础上进行改进，在未知 WPCR 外部需求订单的前提下，需要制定一个稳妥的单周生产计划，在总成本最小的前提下，生产的 WPCR 还要以一定概率满足交货需求。因此，为使总成本最小，可以使用问题二的优化模型，约束条件都是一致的。在使用问题二模型时需要确定好单周的 WOCR 需求，通过将前 30 周的历史数据作为样本空间，可以求出一周内每天以及每周的 WPCR 需求满足一定置信度的单侧置信上限。综合考虑后可将求出的满足交付条件单周 WPCR 需求代入问题二的优化模型，便可求解出问题四的单周生产计划。

8.2 模型建立

由于问题四提出需要以 95%的概率保证正常交付每天的 WPCR 订单，以 85%的概率保障正常交付整周的 WPCR 订单。因此，可以先将前三十周数据作为样本空间，对于一周内某天存在 30 个样本，以此为该天需求的样本空间，设其近似服从正态分布，则可求解某天的置信水平为 0.95 的单侧置信上限：

$$\mathrm{mu_i} = \bar{a_\imath} + \frac{s}{\sqrt{30}} t_\alpha(29)$$

同理，将 30 周数据进行每周 WPCR 需求求和，得到每周的需求样本，设其近似服从正态分布，则可求解出某周的置信水平为 0.85 的单侧置信上限：

$$mu_8 = \overline{a_8} + \frac{s}{\sqrt{30}} t_{\alpha}(29)$$

将得到的结果综合考虑，必须满足必要概率的交货条件，进行取值，得到单周每天 WPCR 需求指标。最后将其与问题二优化模型进行结合：将求得的单周每天 WPCR 需求指标带入问题二模型中的需求矩阵，便可求得满足问题四条件的单周生产计划。

8.3 模型求解

通过 matlab 软件求解得到某天 WPCR 需求单侧置信上限为：

表 9 问题四某天 WPCR 需求单侧置信上限

	周一	周二	周三	周四	周五	周六	周日
WPCR	38.3865	36.2186	39.3496	40.5066	37.9857	34.7961	40.8850

因为生产的 WPCR 产品必须是一个整数，在追求总成本最低的情况下，要使生产数量尽量减小，可将求得的上限进行向下取证：

表 10 问题四某天 WPCR 需求

	周一	周二	周三	周四	周五	周六	周日	求和
WPCR	38	36	39	40	37	34	40	264

同理，求得单周 WPCR 需求单侧置信上限为 263.2977，向下取整为 263。因此，若要达到题目所要求的以 85%概率完成整周的订单，每周至少完成 263 个产品的制作，结合表 9 求和项可知一周内每天需求按表 9 安排，既能保障每天的 WPCR 订单均以 95%以上的概率保证正常交付，又能够以 85%以上的概率保证整周的 WPCR 订单能正常交付。

在得到 WPCR 生产需求后，将其代入问题二的优化模型进行线性规划经典求解，可以得到以下周生产计划：

表 11 问题 4 生产规划

周次/项目	WPCR 外部需求数量	WPCR 组装数量	A 组装数量	B 组装数量	C 组装数量	生产准备费用	库存费用
周一	38	0	249	334	416	630	33763.2
周二	36	83	0	0	500	750	31306.7
周三	39	0	300	370	0	280	2106.7
周四	40	100	0	0	0	420	10212.7
周五	37	0	0	352	0	480	42593.7
周六	34	0	243	0	404	300	1702.5
周日	40	81	0	0	0	740	52657
总和	264	264	792	1056	1320	总成本（元）	177942.5

以上便是我们对目标函数求解得出的最优解，定制好周生产计划表，通过计算得出最终成本为 177942.5 元，包括生产准备费用 3600 元及库存费用 174342.5 元，每日组装数量及每日库存数量均已经列出，如下表：

表 12 问题四生产准备费用

项目/日	周一	周二	周三	周四	周五	周六	周日
WPCR	0	240	0	240	0	0	240
A	120	0	120	0	0	120	0
A1	0	40	0	0	40	0	40
A2	0	60	0	0	60	0	60
A3	0	50	0	0	50	0	50
B	160	0	160	0	160	0	0
B1	0	80	0	80	0	0	80
B2	0	100	0	100	0	0	100
C	180	180	0	0	0	180	0
C1	60	0	0	0	60	0	60
C2	40	0	0	0	40	0	40
C3	70	0	0	0	70	0	70
总费用	630	750	280	420	480	300	740

接问题 2 的条件，该问题在当天需要组装部件仍需提前一天进行生产准备，采购部件同样占据着总成本的较大比例，生产计划每一组件每日库存费用如下：

表 13 问题四每日库存

项目数	周一	周二	周三	周四	周五	周六	周日
WPCR	3	50	11	71	34	0	41
A	249	0	300	0	0	243	0
A1	362	30	400	0	352	352	28
A2	416	501	501	1	1	405	0
A3	0	1800	0	0	1458	0	1494
B	0	2400	0	0	1944	0	1992
B1	0	600	0	0	486	0	498
B2	0	740	0	704	0	0	668
C	0	1480	0	1408	0	0	1336
C1	4000	0	0	0	3232	0	3328
C2	1000	0	0	0	808	0	832
C3	6000	0	0	0	4848	0	4992
总费用	33763.2	31306.7	2106.7	10212.7	42593.7	1702.5	52657

九、模型评价及推广

9.1 模型优点

（1）所建立的优化模型具有可靠的数学基础，综合考虑了各种约束条件，寻得了在一定范围下的最有参数解，保证了参数的正确性与结果的可靠性及合理性。如模型 1 及模型 2 采用线性规划求解，将问题简化，求解简单，使用方便作为一个基础简化模型，其为下一个模型如模型 3、模型 4 打下良好基础，明确给出了物流平衡方程等模型所需满足的最根本的条件，保证了求解结果的正确性。

（2）模型 3 综合考虑了产品的生产批量与检修问题，增添了新的决策变量参数，全面且精确，体现了模型易应用易于改进等特点，结合对模型的检验，使得求解结果的准确度提高。

（3）模型 4 首先对历史订单数据进行拟合，每日订单能保证交付的概率作为了单侧置信区间的置信上限，运用了区间估计对生产计划进行了制定，这种求解思路满足估计量的评选标准如无偏性、有效性、相合性。

9.2 模型缺点

模型的不足之处在于所设约束条件过多，人工计算较为复杂；应用场景较为特殊化，

不能直接用于普遍批量问题的最优求解，如除生产准备费用及当日库存费用外一切费用钧未考虑，部分组件时间忽略未计。其余有关原材料、人力和能量等直接成本作为一个常数均未考虑。

9.3 模型推广

由于时间有限，本模型还要不足之处，但求得结果还是比较合理，均满足了题设条件。Matlab 基于问题求解建模代码的特点在于其代码本身的矩阵化编程风格的完美嵌入，约束条件往往可以不用嵌套循环关键字。代码的简洁也是其得以推广的重要原因之一，模型给出的物流守恒约束条件关系式适用于绝大多数生产批量-库存安排问题。针对不同的问题，只需在本模型基础上，增设约束条件，使用 Matlab 代码求解即可。

十、参考文献

【1】qibbxxt.建模训练案例-3：有瓶颈设备多级生产计划问题中的复杂约束. https://www.ilovematlab.cn/thread-610250-1-1.html(出处: MATLAB 中文论坛) 2022 年 8 月 6 日午 12 时

【2】盛骤，谢式千，潘承毅.概率论与数理统计-第四版.北京.高等教育出版社.2008 年 6 月（2018 年 6 月重印）

【3】司守奎，孙玺菁.数学建模算法与应用-第三版.北京.国防工业出版社.2022 年 2 月重印

附录

Matlab 代码

（附录 1）问题一求解代码：

```
clear;clc;close all
%问题一求解代码
% 以下为所需基本数据
[N,T] = deal(12,7);%N 为产品数量，T 为天数
Demand = [39 36 38 40 37 33 40;zeros(12,7)];%每一天 WPCR 需求量
Limit = [4500 2500 2750 2100 2500 2750 1500]';%生产时间限制
Set = [240 120 160 180 40 60 50 80 100 60 40 70];%生产准备费用
Hold = [5 2 1.5 1.7 5 3 6 4 5 3 2 3];%库存费用
Consume = [0 3 5 5 0 0 0 0 0 0 0 0];%消耗工时
R = full(sparse(2:12,[1 1 1 2 2 2 3 3 4 4 4],[3,4,5,6,8,2,2,4,8,2,12],N,N));%存储不
同组件所需后继组件数量
S = {[],1,1,1,2,2,2,3,3,4,4,4};%用来代表不同组件

% 决策变量如下
x = optimvar('x',N,T,'type','integer','LowerBound',0);%每天生产部件数量
y = optimvar('y',N,T,'type','integer','LowerBound',0,"UpperBound",1);%用来表示是否
需要生产某组件，0 代表不生产，1 代表要生产
Inv = optimvar('Inv',N,T,'type','integer','LowerBound',0);%每种组件在每天的库存数量

prob = optimproblem;%目标函数最小化的优化问题
prob.Objective=sum(Set*y+Hold*Inv);%目标函数

% 约束 1：物流守恒约束
Con1 = optimeq(N, T);%创建一个 N*T 空优化等式数组
for i = 1 : N
  for t = 1 : T
      if t == 1
      Con1(i,t) =x(i,t) - Inv(i,t) == Demand(i,t) + R(i,S{i})*x(S{i},t);%第一天没有
库存量
      else
      Con1(i,t) = Inv(i,t-1) + x(i,t) - Inv(i,t) == Demand(i,t) +
R(i,S{i})*x(S{i},t);%其余时间
      end
  end
end
prob.Constraints.con1 = Con1;
prob.Constraints.con3 = (Consume*x)' <= Limit;%约束二：生产工时限制
prob.Constraints.con4 = x <= 5e4*y; % 约束三：用一个足够大的常数来约束每天的产量
[sol,fvl] = solve(prob)%输出最小成本
```

```
sol.x,sol.y,sol.Inv
```

（附录 2）问题二求解代码：

```
clc,clear
clear;clc;close all
%问题二求解代码
% 以下为所需基本数据
[N,T] = deal(12,7);%N 为产品数量, T 为天数
Demand = [39 36 38 40 37 33 40;zeros(11,7)];%每一天 WPCR 需求量
Limit = [4500 2500 2750 2100 2500 2750 1500]';%生产时间限制
Set = [240 120 160 180 40 60 50 80 100 60 40 70];%生产准备费用
Hold = [5 2 1.5 1.7 5 3 6 4 5 3 2 3];%库存费用
Consume = [0 3 5 5 0 0 0 0 0 0 0 0];%消耗工时
R = full(sparse(2:12,[1 1 1 2 2 2 3 3 4 4 4],[3,4,5,6,8,2,2,4,8,2,12],N,N));%存储不同组件所需后继组件数量
S = {[],1,1,1,2,2,2,3,3,4,4,4};%用来代表不同组件

% 决策变量如下
x = optimvar('x',N,T,'type','integer','LowerBound',0);%每天生产部件数量
y = optimvar('y',N,T,'type','integer','LowerBound',0,"UpperBound",1);%用来表示是否需要生产某组件, 0 代表不生产, 1 代表要生产
Inv = optimvar('Inv',N,T,'type','integer','LowerBound',0);%每种组件在每天的库存数量
z=optimvar('z',N,1,'type','integer','LowerBound',0);
prob = optimproblem;%目标函数最小化的优化问题
prob.Objective=sum(Set*y+Hold*Inv);%目标函数

% 约束 1: 物流守恒约束
Con1 = optimeq(N, T);%创建一个 N*T 空优化等式数组
for i = 1 : N
  for t = 1 : T
      if t == 1
      Con1(i,t) =Inv(i,7)+x(i,t) - Inv(i,t) == Demand(i,t) + R(i,S{i})*x(S{i},t);%周日库存+今日生产-今日库存=今日需求+今日生产后继产品所需数量
      else
      Con1(i,t) = Inv(i,t-1) + x(i,t) - Inv(i,t) == Demand(i,t) + R(i,S{i})*x(S{i},t);%前一天库存+今日生产-今日库存=今日需求+今日生产后继产品所需数量
      end
  end
end
prob.Constraints.con1 = Con1;
prob.Constraints.con3 = (Consume*x)' <= Limit;%约束二: 生产工时限制
```

```
prob.Constraints.con4 = x <= 5e4*y; % 约束三：用一个足够大的常数来约束每天的产量
Con5=optimineq(N,T);%创建一个 N*T 空优化不等式数组

%约束四：对于每天组装所消耗的配件数量不大于前一天的库存数量
for i = 2 : N
  for t = 1 : T
      if t == 1
      Con5(i,t) =R(i,S{i})*x(S{i},t)<= Inv(i,7);%周一消耗不大于星期天的库存
      else
      Con5(i,t) =R(i,S{i})*x(S{i},t)<= Inv(i,t-1);%其他时候的消耗不能够大于前一天的库存
      end
  end
end
prob.Constraints.con5=Con5;
Con6=optimineq(N,1);%创建一个 N*1 空优化不等式数组
s=sum(x,2);%计算每一种产品七天的生产总和
for i=1:N
Con6(i)=Inv(i,7)<=s(i);%约束五：星期天的库存量不大于七天生产数量的总和
end
prob.Constraints.con6=Con6;
[sol,fvl] = solve(prob)%求解并输出最小成本
sol.x,sol.y,sol.Inv
```

（附录 3）问题三求解代码：

```
clc,clear
clear;clc;close all
%问题三求解代码
% 以下为所需基本数据
[N,T] = deal(12,210);%N 为产品数量，T 为天数
%Demand 为 210 天内每一天的需求量
Demand = [39 36 38 40 37 33 40 39 33 37 43 34 30 39 42 36 35 38 36 35 41 38 36 36
48 34 35 39 38 36 40 40 40 34 39 40 30 36 40 34 36 37 41 36 41 41 38 29 43 33 31
40 42 42 30 40 35 36 38 33 35 37 41 43 35 42 37 36 33 39 38 32 41 36 40 31 34 37
37 41 39 38 35 38 38 38 33 42 42 29 33 39 37 44 38 35 36 38 40 39 38 38 37 34 44
35 36 38 39 39 39 39 43 28 39 41 38 30 38 35 37 40 41 40 35 41 36 35 40 41 37 38
36 37 38 39 41 38 37 44 37 37 37 36 39 33 41 39 37 42 37 36 28 43 40 32 35 45 40
34 43 38 36 37 36 40 28 45 38 40 38 36 35 40 42 31 31 44 36 31 36 40 40 36 34 43
35 32 39 33 33 36 41 34 38 40 35 34 37 37 39 36 40 37 41 39 41 36 32
44;zeros(11,210)];%每一天 WPCR 需求量
Limit = [4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
```

```
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500 ...
4500 2500 2750 2100 2500 2750 1500];%生产时间限制
Set = [240 120 160 180 40 60 50 80 100 60 40 70];%生产准备费用
Hold = [5 2 1.5 1.7 5 3 6 4 5 3 2 3];%库存费用
Consume = [0 3 5 5 0 0 0 0 0 0 0 0];%消耗工时
R = full(sparse(2:12,[1 1 1 2 2 2 3 3 4 4 4],[3,4,5,6,8,2,2,4,8,2,12],N,N));%存储不同组件所需后继组件数量
S = {[],1,1,1,2,2,2,3,3,4,4,4};%用来代表不同组件
% 决策变量如下
x = optimvar('x',N,T,'type','integer','LowerBound',0);%每天生产部件数量
y = optimvar('y',N,T,'type','integer','LowerBound',0,"UpperBound",1);%用来表示是否需要生产某组件，0代表不生产，1代表要生产
Inv = optimvar('Inv',N,T,'type','integer','LowerBound',0);%每种组件在每天的库存数量
W = optimvar('W',1,T,'type','integer','LowerBound',0,"UpperBound",1);%表示当天是否需要检修，若为0则表示需要检修，为1则不需要
prob = optimproblem;%目标函数最小化的优化问题
prob.Objective=sum(Set*y+Hold*Inv);%目标函数
```

```
% 约束 1：物流守恒约束（每天组装的数量需要加上前一天的库存量=当天的需求加上当天的库存加上
当天组装所消耗的数量）
Con1 = optimeq(N, T);%创建一个 N*T 空优化等式数组
for i = 1 : N
  for t = 1 : T
      if t == 1 %第一天
      Con1(i,t) =Inv(i,210)+x(i,t) - Inv(i,t) == Demand(i,t) + R(i,S{i})*x(S{i},t);
      else
      Con1(i,t) = Inv(i,t-1) + x(i,t) - Inv(i,t) == Demand(i,t) +
R(i,S{i})*x(S{i},t);%前一天库存+今日生产-今日库存=今日需求+今日生产后继产品所需数量
      end
  end
end
prob.Constraints.con1 = Con1;
Con2=optimineq(N,T);%创建一个 N*1 空优化不等式数组
%约束 2：生产工时限制
for i=1:N
  for t=1:T
      switch W(1,t)
        case 0 %如果当天检修，则之后五天的生产时间限制量会增多
        Con2(i,t+1)=Consume(i)*x(i,t+1)<=1.1*Limit(t+1);
        Con2(i,t+2)=Consume(i)*x(i,t+2)<=1.08*Limit(t+2);
        Con2(i,t+3)=Consume(i)*x(i,t+3)<=1.06*Limit(t+3);
        Con2(i,t+4)=Consume(i)*x(i,t+4)<=1.04*Limit(t+4);
        Con2(i,t+5)=Consume(i)*x(i,t+5)<=1.02*Limit(t+5);
        t=t+5;
        case 1 %当天不检修，则工时限制不变
        Con2(i,t)=Consume(i)*x(i,t)<=Limit(t);
    end
  end
end
prob.Constraints.con2 =Con2;
prob.Constraints.con7 =sum(W)==203;%约束 3：210 天内只能够安排七天检修
for t=1:205
a(t)=W(1,t)+W(1,t+1)+W(1,t+2)+W(1,t+3)+W(1,t+4)+W(1,t+5);
end
prob.Constraints.con8 =a>=5;%约束 4：每两次检修时间至少需要间隔 6 天
prob.Constraints.con4 = x <= 5e4*y; % 约束 5：用一个足够大的常数来约束每天的产量，同时
建立 x 与 y 之间的关系
Con5=optimineq(N,T);%创建一个 N*T 空优化不等式数组

%约束 6：对于每天组装所消耗的配件数量不大于前一天的库存数量
for i = 1 : N
  for t = 1 : T
      if t == 1
```

```
      Con5(i,t) =R(i,S{i})*x(S{i},t)<= Inv(i,210);%第一天消耗不大于第 210 天的库存
      else
      Con5(i,t) =R(i,S{i})*x(S{i},t)<= Inv(i,t-1);%其他时候的消耗不能够大于前一天的库存
      end
  end
end
prob.Constraints.con5=Con5;
Con6=optimineq(N,1);%创建一个 N*1 空优化不等式数组
s=sum(x,2);%计算每一种产品七天的生产总和
for i=1:N
Con6(i)=Inv(i,210)<=s(i);%约束 7：第 210 的库存量不大于 210 天以来生产数量的总和
end
prob.Constraints.con6=Con6;
prob.Constraints.con11=sum(x,1)<=5e4*W;%约束 8：建立 x 与 W 之间的关系
[sol,fvl] = solve(prob)%求解输出最小成本
sol.x,sol.y,sol.Inv,sol.W
```

（附录 4）问题四求解代码：

```
clear;clc;close all
%问题四求解代码
a=[39 39 42 38 38 40 41 33 35 43 38 37 38 39 40 35 43 35 36 37 37 39 40 38 38 31 40 33 35 37;
36 33 36 36 36 30 36 31 36 35 32 37 38 37 39 36 28 37 35 38 37 37 32 36 40 31 36 33 34 41;
38 37 35 36 40 36 41 40 38 42 41 41 33 44 38 38 39 40 40 39 37 42 35 37 38 44 34 36 37 39;
40 43 38 48 40 40 41 42 33 37 36 39 42 38 38 39 41 41 41 41 36 37 45 36 36 36 43 41 37 41;
37 34 36 34 40 34 38 42 35 36 40 38 42 35 37 39 38 40 37 38 39 36 40 40 35 31 35 34 39 36;
33 30 35 35 34 36 29 30 37 33 31 35 29 36 34 39 30 35 38 37 33 28 34 28 40 36 32 38 36 32;
40 39 41 39 39 37 43 40 41 39 34 38 33 38 44 39 38 41 36 44 41 43 43 45 42 40 39 40 40 44;
263 255 263 266 267 253 269 258 255 265 252 265 255 267 270 265 257 269 263 274 260 262 269 260 269 249 259 255 258 270];%30 周历史 WPCR 订单数据
%可选择显示需求图像
% subplot(2,4,1); plot(a);
% subplot(2,4,2);plot(a1);
% subplot(2,4,3);plot(a2);
% subplot(2,4,4);plot(a3);
```

```
% subplot(2,4,5);plot(a4);
% subplot(2,4,6);plot(a5);
% subplot(2,4,7);plot(a6);
% subplot(2,4,8);plot(a7);

%计算周一到周日
% 每天置信水平为 0.95 的单侧置信上限
for i=1:7
alpha=0.05;%显著性水平
ta=tinv(1-alpha,29);%t 分布表中随测点数(29)和置信度(alpha)而变的系数
xb=mean(a(i,:));%均值
s=std(a(i,:));%标准差
n=length(a(i,:));%样本数量
mu(i,1)=xb+s/sqrt(n)*ta;%计算单侧置信上限
end
%计算整周需求的置信水平为 0.85 的单侧置信上限
alpha=0.15;
ta=tinv(1-alpha,29);
xb=mean(a(8,:));
s=std(a(8,:));
n=length(a(8,:));
mu(8,1)=xb+s/sqrt(n)*ta;
mu=floor(mu(:,1))%向下取整
mu(8,:)=[];%删除整周的单侧置信上限，为与下面 Demand 结合准备

% 以下为所需基本数据
[N,T] = deal(12,7);%N 为产品数量，T 为天数
Demand = [mu';zeros(11,7)];%每一天 WPCR 需求量
Limit = [4500 2500 2750 2100 2500 2750 1500]';%生产时间限制
Set = [240 120 160 180 40 60 50 80 100 60 40 70];%生产准备费用
Hold = [5 2 1.5 1.7 5 3 6 4 5 3 2 3];%库存费用
Consume = [0 3 5 5 0 0 0 0 0 0 0 0];%消耗工时
R = full(sparse(2:12,[1 1 1 2 2 2 3 3 4 4 4],[3,4,5,6,8,2,2,4,8,2,12],N,N));%存储不同组件所需后继组件数量
S = {[],1,1,1,2,2,2,3,3,4,4,4};%用来代表不同组件

% 决策变量如下
x = optimvar('x',N,T,'type','integer','LowerBound',0);%每天生产部件数量
y = optimvar('y',N,T,'type','integer','LowerBound',0,"UpperBound",1);%用来表示是否需要生产某组件，0 代表不生产，1 代表要生产
Inv = optimvar('Inv',N,T,'type','integer','LowerBound',0);%每种组件在每天的库存数量
prob = optimproblem;%目标函数最小化的优化问题
prob.Objective=sum(Set*y+Hold*Inv);%目标函数
```

```
% 约束 1：物流守恒约束
Con1 = optimeq(N, T);%创建一个 N*T 空优化等式数组
for i = 1 : N %从 1-12，各个组件循环
  for t = 1 : T %从周一到周日循环
      if t == 1 %周一
      Con1(i,t) =Inv(i,7)+x(i,t) - Inv(i,t) == Demand(i,t) + R(i,S{i})*x(S{i},t);%
周日库存+今日生产-今日库存=今日需求+今日生产后继产品所需数量
      else
      Con1(i,t) = Inv(i,t-1) + x(i,t) - Inv(i,t) == Demand(i,t) +
R(i,S{i})*x(S{i},t);%前一天库存+今日生产-今日库存=今日需求+今日生产后继产品所需数量
      end
  end
end
prob.Constraints.con1 = Con1;%引用 con1
prob.Constraints.con3 = (Consume*x)' <= Limit;%约束二：生产工时限制
prob.Constraints.con4 = x <= 5e4*y; % 约束三：用一个足够大的常数来约束每天的产量
Con5=optimineq(N,T);%创建一个 N*T 空优化不等式数组

%约束四：对于每天组装所消耗的配件数量不大于前一天的库存数量
for i = 2 : N
  for t = 1 : T
      if t == 1
      Con5(i,t) =R(i,S{i})*x(S{i},t)<= Inv(i,7);%周一消耗不大于星期天的库存
      else
      Con5(i,t) =R(i,S{i})*x(S{i},t)<= Inv(i,t-1);%其他时候的消耗不能够大于前一天的库
存
      end
  end
end
prob.Constraints.con5=Con5;%引用 Con5
Con6=optimineq(N,1);%创建一个 N*1 空优化不等式数组
s=sum(x,2);%计算每一种产品七天的生产总和
for i=1:N %从周一到周日
Con6(i)=Inv(i,7)<=s(i);%约束五：星期天的库存量不大于七天生产数量的总和
end
prob.Constraints.con6=Con6;%引用 Con6
[sol,fvl] = solve(prob)%求解并输出最小成本
sol.x,sol.y,sol.Inv%显示各决策变量
```